Disclosure IL8-2000-0022

Prepared for and/or by an IBM Attorney - IBM Confidential

Created By Dori Koriat On 23/10/2000 04:08:34 AM EST
Last Modified By wpts1 wpts1 On 07/01/2005 07:19:11 PM EST
Archived on 02/02/2002

Required fields are marked with the asterisk (*) and must be filled in to complete the form.

***Title of disclosure (in English)**
IMPROVED METHOD TO DETECT END OF RDMA TRANSFER

Summary

Status	Final Decision (FIle)
Final deadline	
Final deadline reason	
Docket family	IL9-2000-0043
* Processing location	Israel
* Functional area	(11) Storage & Systems Technology
Attorney/Patent professional	Jules D Williams/Zurich/IBM
IDT team	
Submitted date	27/02/2000
* Owning division	HRL
Incentive program	
Lab	
* Technology code	
Patent value tool (PVT) score	

Inventors with a Blue Pages entry

Inventors: Kalman Meth/Haifa/IBM, Julian Satran/Haifa/IBM

Inventor Name	Inventor Serial	Div/Dept	Inventor Phone	Manager Name
Meth, Kalman (Zvi)	429209	N/A/403	N/A	Azagury, Alan
Satran, Julian	646704	N/A/403	N/A	Rodeh, Michael

> denotes primary contact

Inventors without a Blue Pages entry

IDT Selection

Main Idea

To view the Main Idea of this disclosure, open the "Main Idea" document from the view

***Critical Questions (Questions 1-9 must be answered in English)**

***Patent Value Tool (Optional - this may be used by the inventor and attorney to assist with the evaluatior**

Final Decision

Post Disclosure Text & Drawings

Form Revised (05/28/03)

Exhibit A

Main Idea for Disclosure IL 8-2000-0022

Prepared for and/or by an IBM Attorney - IBM Confidential

Archived On 02/11/2000 08:02:34 AM

Title of disclosure (in English)
Improved Method to Detect End of RDMA Transfer

Main Idea

1. Describe your invention, stating the problem solved (if appropriate), and indicating the advantages of using the invention.

rmdaend.lwp

2. How does the invention solve the problem or achieve an advantage,(a description of "the invention", including figures inline as appropriate)?

3. If the same advantage or problem has been identified by others (inside/outside IBM), how have those others solved it and does your solution differ and why is it better?

4. If the invention is implemented in a product or prototype, include technical details, purpose, disclosure details to others and the date of that implementation.

Invention Disclosure:

Improved Method to Detect End of RDMA Transfer

Authors:
Julian Satran
Kalman Meth
27 Feb 2000 - first draft

Statement of Problem:

RDMA refers to a Remote DMA (Direct Memory Access) feature that is provided on some communications infrastructures. The sender of data specifies, in a form understood by the receiver, where the data should be placed at the receiving end; the application on the receiving end might then place the data without having to examine a complex context, or might even delegate the data placement to specialized hardware. When data has been successfully delivered into the receiver's buffers, the receiver must be notified of the completed transfer (usually be some kind of interrupt mechanism).

There is sometimes a problem, however, to determine when a data transfer has completed, especially if a large data transfer (which we will call a transaction) has been broken into several smaller data transfers (which we will call packets). It would be desirable to inform the receiver that the entire transaction (large data transfer) has been completed, without interrupting (disturbing) the receiver when only some packet (partial data transfer) has completed. An RDMA engine may know how much data has been transferred on each packet (small data transfer), and it may know how much data makes up the entire transaction. The RDMA engine would then have to keep track of how much data has arrived for each pending transaction (large data transfer), and would generate an interrupt when it has received the total number of bytes that were specified for a particular transaction (after having received some number of packets).

The problem is compounded by allowing the transaction (large data transfer) to be broken into several smaller pieces (packets) that may traverse different network fabrics. In this case, no single RDMA engine on the receiving end receives all of the data for a particular transaction (large data transfer), and therefore no single RDMA engine can know when the transaction (large data transfer) has completed. In current state-of-the-art RDMA proposals/implementations, this is solved by generating an interrupt or callback for each packet (small data transfer) on each of the RDMA engines, and computing the total data delivered for the transaction in software. This solution has the undesirable condition that it results in an interrupt being generated for each packet (small data transfer). The receiver is interested in knowing when the entire transaction (large data transfer) has completed, and all of the extra interrupts/callbacks for the small data transfers consume resources that could otherwise be used for other purposes.

It is therefore desirable to have a solution to this problem that minimizes the number of interrupts in determining when a transaction (large data transfer) using RDMA has completed.

Claim:

We propose a method that reduces the number of interrupts. Each RDMA engine that receives some data for a transaction will produce one (and only one) interrupt/callback for that transaction. Any RDMA engine that did not receive any data for a transaction will not produce any interrupts for that transaction.

Solution/Embodiment:

The sender may send parts of data (packets) of the transaction over several network fabrics/connections. When the sender has sent the last packet of data through a particular network, the sender will mark the end-of data through a marker that can be a flag (in the message header) that indicates that this is the last piece of data being sent on this network connection/fabric for the particular transaction or an specially formatted message (e.g. an empty RDMA packet). If the sender finished sending out data for a transaction, but it had sent data earlier over a network without marking the last packet sent on that network, the sender must send a specially formatted message (e.g.. an empty RDMA packet) that marks it as the last packet being sent over that network for the particular transaction. Each receiver thus knows which packet is the last packet it will receive for a particular transaction. Upon receiving this last packet, the RDMA engine generates an interrupt/callback, informing the receiver how much data it has received on its network connection/fabric for the particular transaction. The receiver then keeps track of the sum of data that arrived on each of the network connections/fabrics that reported data received for the particular transaction. When the total number of bytes for the transaction has been received via the various RDMA engines, the receiver knows that the transaction (large data transfer) has completed. Any RDMA engine that did not process any packets for a transaction will not have generated an interrupt for that transaction. Any RDMA engine that did process packets for a transaction will have generated a single interrupt for the transaction after it had processed all of the packets that are to arrive on its network connection/fabric. We have thus reduced the number of interrupts needed to determine when a split RDMA transfer has been completed.

A variant of this technique may involve the information sender to inform the receiver about the connections on which it has sent data enabling him to cross-check the validity of the receive-counts.